2010

Hiding secrete data in compressed images using histogram analysis

Farhad Keissarian

University of Wollongong in Dubai, farhadk@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/dubaipapers>

Recommended Citation

Keissarian, Farhad: Hiding secrete data in compressed images using histogram analysis 2010.
<https://ro.uow.edu.au/dubaipapers/255>

Hiding Secrete Data in Compressed Images Using Histogram Analysis

Farhad Keissarian

University of Wollongong in Dubai
farhadkeissarian@uowdubai.ac.ae

Abstract—In this paper, we present a data hiding scheme that embeds the secret data into the compression codes of the host image. The compression codes are associated with the visual patterns appearing in image blocks and are computed through a histogram analysis of residual blocks of the host image. The hiding capacity for each compressed block is not fixed and varies with its block type. Blocks with low visual activity hide more data than those with high visual activity. Experimental results confirm that the proposed technique can provide high data capacity within the compression file with acceptable image quality of the stego-images.

Keywords- data hiding; image compression; block pattern; block histogram.

I. INTRODUCTION

Protection of the transmitted data over the Internet from being intercepted or tampered with has become an increasingly important issue. There are two kinds of protection techniques, cryptography and data hiding. In the former technique, the secret data are encrypted by the sender to be transferred into a set of meaningless data. Only the legal receiver has the means to decode the meaningless data. The data hiding techniques, on the other hand, embed the secret data into a digital media called the cover image to become a stego-image. Because the stego-image is often imperceptible, it arouses less attention from the malicious attackers than data encryption [1].

Data hiding techniques have been roughly classified into three categories: the spatial-domain manner, the frequency-domain manner and the compression-domain manner. For the spatial-domain manner, secret data are mixed directly into the distributed pixels. The least significant bit (LSB) [2] is the general approach to hide the secret information into the LSBs of each pixel of a cover image directly owing to their lower distortion in image quality. For the frequency domain manner, the cover image must first be transformed into frequency coefficients by using a frequency-oriented mechanism such as discrete wavelet transformation (DWT) [3]. Later, the secret data are combined with the relative coefficients in the frequency-form image. For the compression-domain manner, the secret data are embedded into the compression codes. In recent years, compression codes generated by well-known image compressions such as vector quantization (VQ), block truncation coding (BTC) and similar methods have been used for data hiding to extend the variety of cover images.

The method proposed in [4], applies side match vector quantization (SMVQ) with the concept of prediction to

propose an adaptive data hiding scheme. In addition to the VQ compression domain, Chuang *et al* [5] proposed a data embedding scheme based on BTC for gray scale images. In their scheme, they predefined a threshold to classify the type of each BTC-encoded block as smooth or complex. Subsequently, they embedded the secret data into the bitmap of the smooth BTC-encoded blocks.

In this paper, we develop an image hiding scheme that can hide the secret data into compression codes of the host image, generated by the image compression technique that we reported earlier in [6]. The compression codes are associated with the visual patterns appearing in image blocks and are computed through a histogram analysis of residual blocks of the host image. The rest of the paper is organized into four sections. The concept of the proposed compression algorithm is introduced in Section 2. In section 3, the proposed hiding scheme is presented. Experimental results are given in Section 4.

II. IMAGE COMPRESSION ALGORITHM

In the proposed image compression algorithm in [6], an image is block coded according to the type of individual blocks. A novel classifier, which is designed based on the histogram analysis of residual blocks, is employed to classify the blocks according to their level of visual activity. The classifier places each block into one of the two categories of uniform or edge block. A uniform block is coded by the block mean, whereas an edge block is coded by a set of parameters associated with the pattern appearing inside the block. Like the original BTC algorithm [7], our method encodes an edge block by initially computing two gray values and constructing a bit-map. However, in the proposed method the computation of the gray values, namely the low and high representative intensities are carried out through analysis of the block residuals' histogram. Moreover, instead of transmitting the two gray values, their average and difference will be sent to decoder. Finally, instead of transmitting the whole bit-map for the processed edge block, an optimum bit-pattern is selected from a set of pre-defined patterns, and its index will be transmitted. The use of these parameters at the receiver reduces the cost of reconstruction significantly and exploits the efficiency of the proposed technique. A brief description of the block classifier and the coding scheme are given in the next two sub-sections.

A. Block Classifier

A novel histogram-based classification scheme has been developed for classifying the image blocks [6]. The method

operates based on the distribution of the block residuals and classifies block either as a low-detail (uniform) or as a high-detail (edge) block. The classifier employs the block residuals and classifies the block according to their histogram. The classification is carried out through a peak detection method on the histogram. A brief description of the classifier is as follows:

Each block of 4x4 pixels is converted into a residual block by subtracting the sample mean from the original pixels. The residual samples are less correlated than the original samples within a block. Here, two of the most important local characteristics of the image block are considered: *central tendency*, represented by the mean value and the *dispersion* of the block samples about the mean, which is represented by the residual values. The challenge here is to analyze the dispersion of the residual values about the mean. One way of achieving this is to sort the histogram of the block residual samples. As the neighboring pixels in the original block are highly correlated, the residual samples will tend to concentrate around zero. One can then quantize the residual samples prior to forming the histogram. The histogram of the quantized residuals may then be formed and analyzed by simply detecting its peaks.

Based on the distribution of the residual samples within the test images, we choose to apply a coarse quantization, in particular a 15-level non-uniform quantizer. We now define q_j as the output of the quantizer with index j , as shown in “Fig. 1”. The histogram of the quantized values $h(q_j)$ may then be formed to provide the occurrence of q_j . The quantized residual histogram (QRH) is then analyzed by simply detecting its peaks. According to the number of detected distinct peaks on the histogram, image blocks can be placed into two major categories of uniform and edge blocks. A histogram with a unique peak at its centre (unimodal histogram) identifies a uniform block. Whereas, the existence of two distinct peaks implies that the processed block is an edge block. “Fig. 2” and “Fig. 3” show examples of both types of blocks.

A peak on the histogram indicates a high score of residual values; therefore it is fair to conclude that there is a considerable number of pixels that have the same dispersion about the block mean. This, in turn will lead us to conclude that the gray level values of these pixels are very close to one another. Hence, this group of pixels can be represented by a single gray value. In this analysis, a distinct peak on the QRH of the processed block represents a gray value X_j , given as :

$$X_j = X_{mean} + q_j \quad (1)$$

j	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
q_j	-67	-52	-39	-28	-17	-10	-5	0	5	10	17	28	39	52	67

Figure 1. The quantizer output q_j with index j

Figure 2. (a) Original uniform block; sample mean = 181, (b) Residuals (c) quantized residuals, (d) quantizer's indexes, (e) block QRH

Where X_{mean} is the block mean. For a uniform block, since the single peak occurs at the center of the histogram, where $q_j = 0$, then from Eq.1 the representative intensity X_j will be the same as the block mean. For an edge block, the two peaks of the QRH, which are positioned on the left and right hand side of the centre ($j=0$) represent the low representative intensity X_L and the high representative intensity X_H , respectively. If the two peaks are positioned at indexes j' and j'' , the two representative intensities are calculated as:

$$\begin{aligned} X_L &= X_{mean} + q_{j'} \\ X_H &= X_{mean} + q_{j''} \end{aligned} \quad (2)$$

In “Fig. 3”, $X_{mean} = 125$, and X_L and X_H are computed using Eq. 2 ; $X_L = 125 + (-39) = 86$ and $X_H = 125 + 28 = 153$.

Figure 3. (a) Original edge block: sample mean = 125 (b) Residuals (c) quantized residuals, (d) quantizer's indexes, (e) QRH

B. Coding of Image Blocks

Once the image blocks have been classified, the coder, switches between a one-level (uniform block) and a bi-level(edge block) representation. A uniform block is encoded by transmitting the block mean plus an indicator to inform the decoder that the block is uniform. By forcibly clustering all pixels in an edge block into two groups, a bi-level approximation of the block is obtained. The clustering partitions a block W into two sets of pixels, W_0 and W_1 , such that $W = W_0 \cup W_1$ and $W_0 \cap W_1 = \Phi$. The clustering is carried out by marking the pixels of set W_0 and W_1 by '0' and '1', respectively. Thus the clustering can be represented as a bit-pattern, $B = \{b_1, b_2, \dots, b_{16} | b_i \in (0,1)\}$. By selecting the block mean as a threshold, the bit-pattern can be generated as :

$$b_i = \begin{cases} 1 & \text{if } x_i > X_{mean} \\ 0 & \text{if } x_i \leq X_{mean} \end{cases} \quad (3)$$

where, $x_i \in W$ are the intensities of the pixels of the edge block. It is noted that, X_L and X_H are the representative intensities of the set W_0 and W_1 , respectively.

Figure 4: Set of 32 pre-defined patterns.

Like the original BTC, an edge block can be coded by transmitting the representative intensities and the bit-pattern. However, in our method, we transmit the average, M and difference, l of the representative intensities, defined by :

$$M = \frac{X_H + X_L}{2} \quad (4)$$

$$l = \frac{X_H - X_L}{2}$$

The values M and l represent the low and high frequency components, respectively. It is evident from eq.3 that $X_H = M + l$ and $X_L = M - l$. During the reconstruction, the coded block can be constructed by :

$$b_i = \begin{cases} M + l & \text{if } b_i \in W_1 \\ M - l & \text{if } b_i \in W_0 \end{cases} \quad (5)$$

It should be noted that for a uniform block, since both representative intensities are the same as the block mean, therefore, $M = X_{mean}$ and $l = 0$. Instead of transmitting the whole bit-pattern of an edge block, further bit reduction can be achieved by finding the best match for the block bit pattern from a set of pre-defined patterns, P_k , $k = 0, 1, 2, \dots, N$. A set of 32 patterns shown in "Fig. 4", which preserve the location and polarity of edges in four major directions and their complements making $N=64$ is used in our method. The pattern matching stage is carried out by performing a logical exclusive NOR operation on the block bit-pattern and each pattern from the set to calculate a matching score, ms , given as :

$$ms = \sum_{i=0}^3 \sum_{j=0}^3 (P_{ij} \oplus b_{ij}) \quad (6)$$

The pattern with the highest ms is selected and its index k will be transmitted. Since, the proposed method sends k instead of the whole block bit-pattern, only $\log_2^{64} = 6$ bits are transmitted. Each image block is therefore encoded by generating a triple (M, l, k) . It should be also noted that, since for a uniform block, no pattern index is transmitted, therefore the compressed code for such a block is the pair (M, l) , where $l = 0$. The value M in the triple (M, l, k) , can be coded by 8 bits, whereas coding l requires only 6 bits, as its standard deviation is smaller than that of X_L and X_H . Therefore, the compression code of an edge block requires $20=(8+6+6)$ bits to be transmitted. For a uniform block, the number of bits required to code the pair $(M, 0)$ is $9=8+1$.

III. PROPOSED DATA HIDING SCHEME

The proposed hiding scheme embeds the secret data into the compression codes of the host image, generated by the proposed compression algorithm, described in the previous

section. The secret data is a random bit stream of '0' and '1'. For the sake of data security, secret data should be encrypted by a secret key before being embedded. After embedding secret data into the compression codes, one can use the decompression procedure to obtain the stego-image. The data embedding and data extraction phases are described in the following sub-sections.

A. Data Embedding Phase

In this phase, the encoder receives the secret bits to hide them in the compression codes of uniform and edge blocks. The hiding capacity for each compressed block is not fixed and varies with its block type. For a uniform block, M and the bit-pattern of the block are used for data embedding. Whereas, an edge blocks uses M and l for hiding the secret bits. Since, the pixel intensities in uniform blocks are close to their neighboring pixels, even though the bits in their bit-patterns are changed, the reconstructed pixel value is still close to its original one. Hence, all the bits in the bit-pattern can be replaced by secret bits.

In the proposed compression algorithm, no bit-pattern or pattern index is transmitted while coding a uniform block. This significantly contributes to a high compression ratio. However, in the proposed data hiding method, the compression code of a uniform block is enhanced to include the block bit-pattern. The enhanced compression code of a uniform block is then the triple $(M, 0, B)$, where B is the bit-pattern of the block. This results in significantly enlarging the hiding capacity at the cost of lower compression ratio. Although, the bit-pattern is transmitted for a uniform block, but the reconstruction is still carried out by replacing all the elements of the bit-pattern by the block mean. Therefore, the quality of the reconstructed uniform block will be the same, if the bit-pattern is not transmitted.

In the proposed method, a uniform block has the capacity of embedding sixteen bits into its bit-pattern and two bits in its M value. To further enlarge the hiding capacity of the host image while retaining good image quality, the proposed scheme also hides two extra bits in each of M and l values of an edge block. The modification of M and l for embedding the secret bits is as follows.

The encoder receives two secret bits and then transform these two bits into a secret decimal digit sd . Next, the encoder uses the M and l values to compute F given as :

$$\begin{aligned} F_M &= M \bmod 5 \\ F_l &= l \bmod 5 \end{aligned} \quad (8)$$

Let sd_M and sd_l denote the secret digits to be hidden in M and l , respectively. The modified values M' and l' are then computed as :

$$\begin{aligned} M' &= M + (sd_M - F_M) \\ l' &= l + (sd_l - F_l) \end{aligned} \quad (9)$$

From Eq. 9, it is evident that no modification is needed if a secret digit is equal to the calculated value F . The embedding phase of the proposed method follows the steps listed below :

Step 1: Obtain the triple compression code for the processed block. If the block is uniform, go to step3, otherwise go to step 2.

Step 2: Embed 2 bits in M to get M' and 2 bits in l to get l' . Transmit the triple (M', l', k) . Go to step 4

Step 3: Embed 2 bits in M to get M' and 16 bits in B to get B' . Transmit the triple $(M', 0, B')$. Go to step 4.

Step 4: Repeat step 1, until the entire bit stream is embedded.

Step5 : Decrypt the extracted information to obtain the original secret data

B. Extraction Phase

The data extraction phase is similar to the data embedding phase. The decoder receives the triple code for each block. If the block is uniform, 2 bits are retrieved from M' and 16 bits from the transmitted bit-pattern B' . On the other hand, if the received code is for an edge block, 2 bits are extracted from M' and another 2 bits are extracted from l' . The secret digit sd can be extracted by computing $sd_M = M' \bmod 5$ and $sd_l = l' \bmod 5$. The data extracting process is repeatedly executed until all the secret bits are retrieved. The extracting phase of the proposed method follows the steps listed below :

Step 1: Obtain the triple compressed code for the block. Go to step 2.

Step 2: If $l = 0$, the triple is $(M', 0, B')$. Extract 2 bits from M' and 16 bits from B' , and go to step 4, otherwise go to step 3.

Step 3: The triple is (M', l', k) Extract 2 bits from M' and 2 bits from l' . Go to step 4

Step 4 : Repeat step 1, until all the secret bits are retrieved.

IV. SIMULATION RESULTS

We have evaluated the performance of the proposed coding scheme through a computer simulation on two gray-level images 'Lena' and 'Pepper', shown in "Fig.5a" and "Fig. 6a". These images are 8 bits per pixel and 512 x 512 pixels in size. The simulation platform is Microsoft Windows XP, Pentium III, and the proposed scheme is implemented using Matlab. Three performance matrices are used to measure the performance of the proposed compression and hiding schemes : Hiding capacity (CE), compression ratio (bpp), and image quality (PSNR). Tables 1 and 2 show the compression results, the hiding capacity, HC and the image quality (PSNR) for the cover images 'Lena' and 'Pepper'.

TABLE I

Image	P(u)	CR (bpp)	TC (Kbyte)	HC (Kbyte)	PSNR (dB)
Lena	63%	1.46	46.5	26.76	31.28
Pepper	72%	1.48	47.2	28.16	31.34

P(u): Population of uniform blocks

CR : Compression Ratio,
TC : Transmitted Code
HC : Hiding Capacity

From Table I, it can be observed that more than half of the transmitted code is used to embed data, yet keeping a satisfactory image quality. To further investigate the performance of the proposed scheme, we compare the stego-image quality and the hiding capacity of the proposed scheme with Chuang *et al.*'s scheme for a threshold of $T=15$. The results are shown in Table II. From Table II, we observe that the proposed method outperforms Chuang *et al.*'s scheme in the hiding capacity by 15%, with a very little distortion in the quality.

TABLE II

Image	Method	PSNR (dB)	Hiding Capacity (Kbyte)	Compression Ratio (bpp)
Lena	Chuang <i>et al.</i>	31.40	23.22	2.0
	Proposed	31.28	26.76	1.46
Pepper	Chuang <i>et al.</i>	31.45	24.74	2.0
	Proposed	31.34	28.16	1.48

From Table II, we observe that the proposed method outperforms Chuang *et al.*'s scheme in the hiding capacity by 15%, with a very little distortion in the quality. Since the

REFERENCES

- [1] uuu Petitcolas, F.A.P., Anderson, R.J., and Kuhn, M.G.: 'Information hiding – a survey', Proc. IEEE, 1999, vol. 87, (7), pp. 1062–1078
- [2] J. CK. Chan and LM. Cheng, "Hiding data in images by simple LSB Substitution", Pattern Recog., vol. 37(3), pp. 469–474, 2004
- [3] P. Bao and X. Ma, "Image adaptive watermarking using wavelet domain singular value decomposition", IEEE Trans. Circuits Syst. Video Technol., 2005, vol. 15, (1), pp. 96–102
- [4] SC. Shie, SD. Lin and CM. Fang, "Adaptive data hiding based on SMVQ prediction", IEICE Trans Inform Syst., vol. E89-D(1), pp. 358–362
- [5] JC. Chuang and CC. Chang, "Using a simple and fast image compression algorithm to hide secret information". International Journal of Computes and Applications, vol.28 (4), pp. 329-333, 2006

Figure 5. The compression and embedding results for the Image of "Lena": a) Original image, b) Compressed stego-image: HC=26.76 Kb, at 1.46 bpp, c) Compressed stego-image: HC=5.5 Kb, at 0.81 bpp.

compression algorithm described in section II, does not transmit the bit-pattern of a uniform block B , we also considered hiding data into the compression code of a uniform block without transmitting the block bit-pattern, B . As mentioned earlier, since a uniform block is reconstructed by replacing all the block pixels by the block mean, the quality of reconstructed block will be the same in either cases. Table 3 shows the results for this case.

TABLE III

Image	P(u)	CR (bpp)	TC (Kbyte)	HC (Kbyte)	PSNR (dB)
Lena	63%	0.81	25.9	5.5	31.28
Pepper	72%	0.75	24.2	5.1	31.34

By comparing the results in Table I and Table II, we note that non-transmission of bit-patterns of uniform blocks results in doubling the compression ratio, at the cost of dramatic reduction of hiding capacity by 80%. Depending on the amount of secrete data, one can select the suitable method. "Fig. 5" and "Fig. 6" show the original test images and the compressed stego-images for both cases.

- [6] F Keissarian, "Novel quad-tree predictive image coding technique using pattern-based classification, Proc. SPIE, Visual Communications and Image Processing (VCIP-2003), vol. 5150, pp. 1481-1490, June 2003, Lugano, Switzerland.
- [7] E.J. Delp and O.R. Mitchell, "Image compression using block truncation coding, IEEE Trans. Commun., vol. 27, pp. 1335-1342, 1979.

Figure 6. The compression and embedding results for the Image of "Pepper": a) Original image, b) Compressed stego-image: HC=28.16 Kb, at 1.48 bpp, c) Compressed stego-image: HC=5.1 Kb, at 0.75 bpp.